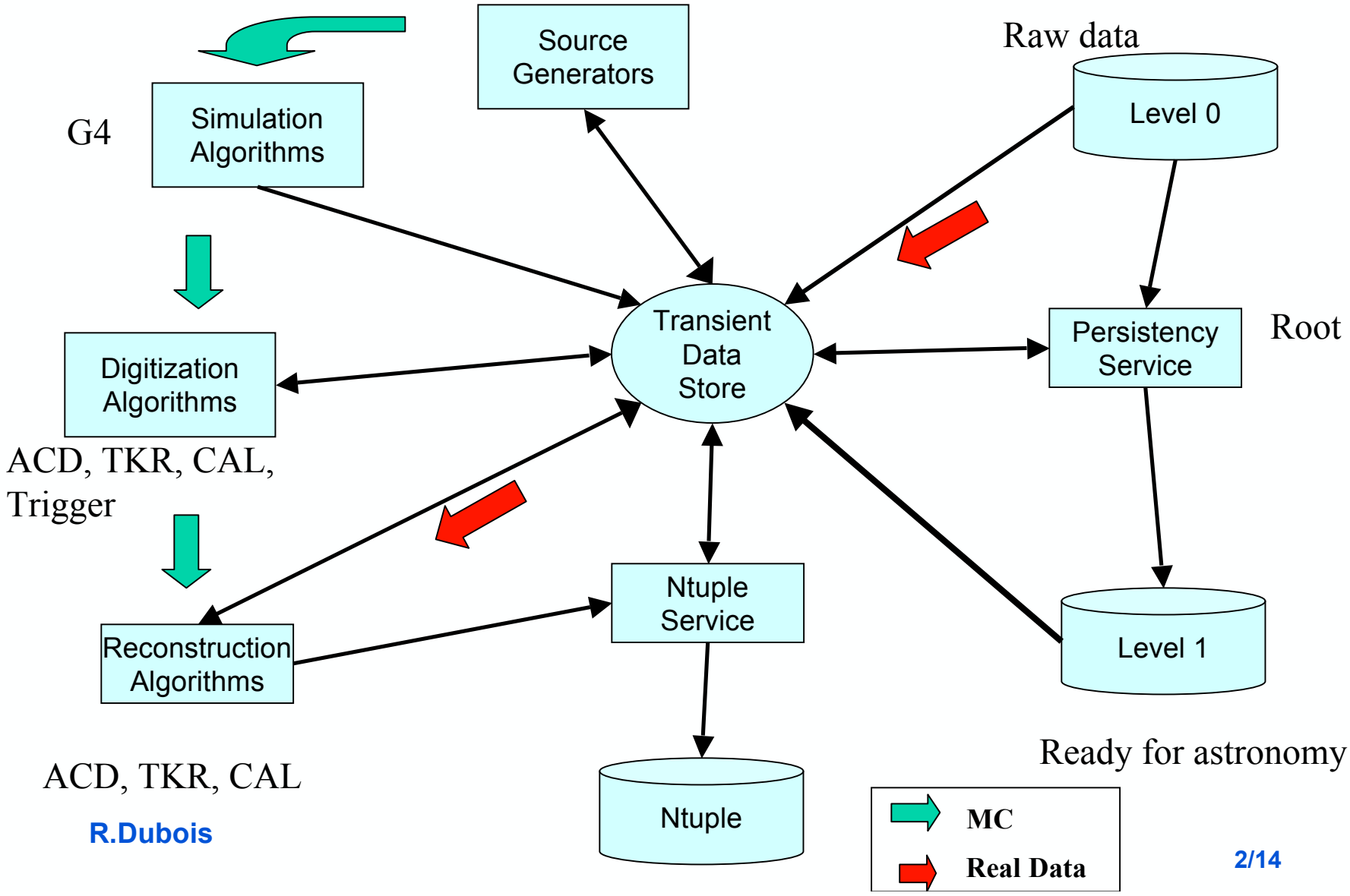
Running the Software

- What
- Where
- When
- How

http://www-glast.slac.stanford.edu/software

Data flow in Gleam



Sim/Recon Developers Toolset

Root, IDL – analysis

applications

Reconstruction
TkrRecon, CalRecon, AcdRecon

simulation package
GEANT4

xml – geometry, parameters

Root – object I/O

Gaudi – code framework

VC++ – Windows IDE
gnu tools - Linux

vcmt – Windows, linux gui

CMT – package version management

ssh – secure cvs access

cvs – file version management

utilities

Sim/Recon Users Toolset

applications

Root, IDL – analysis

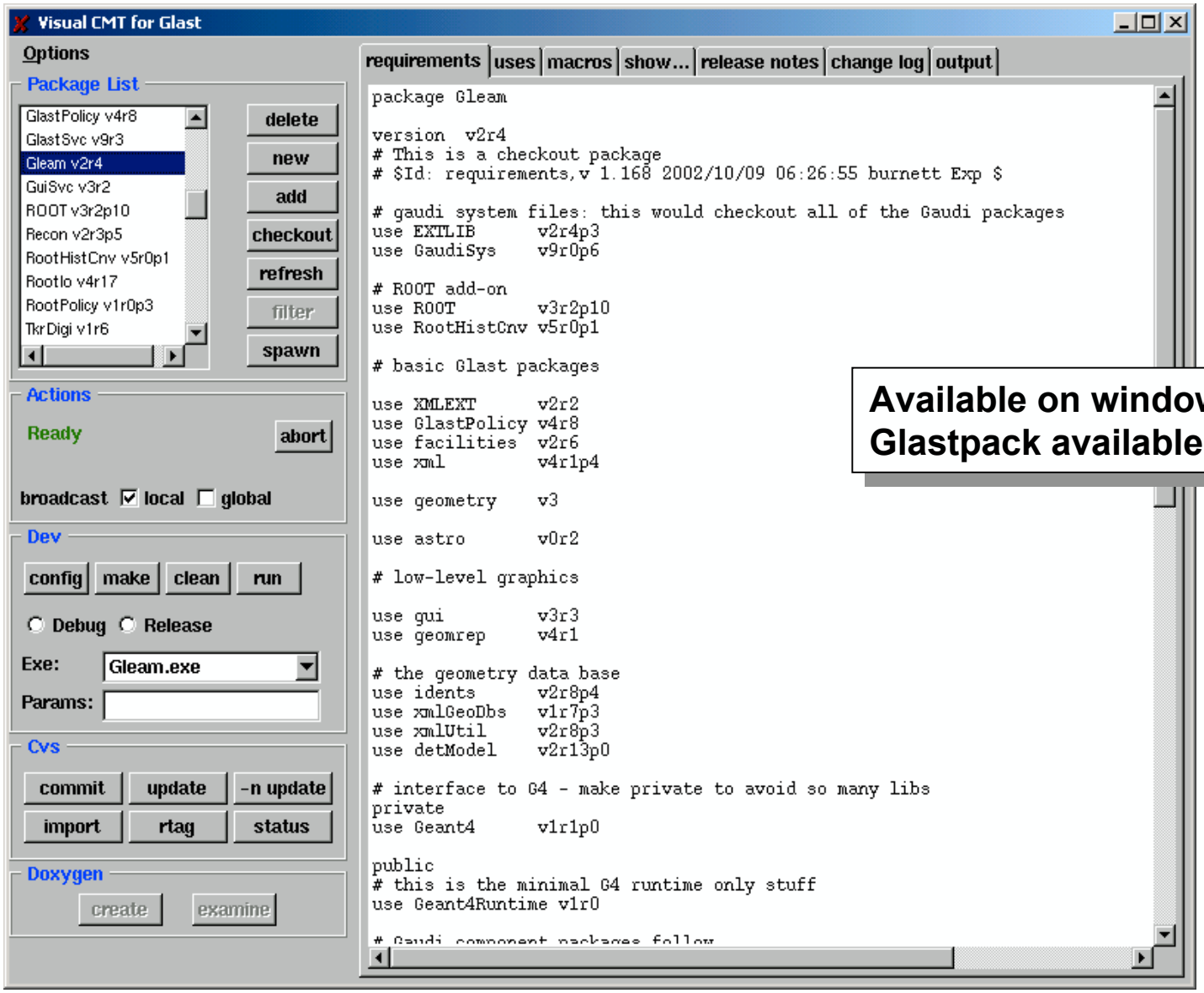
Gleam “binary” distribution – Windows, Linux

vcmt – Windows, linux gui

CMT – package version management

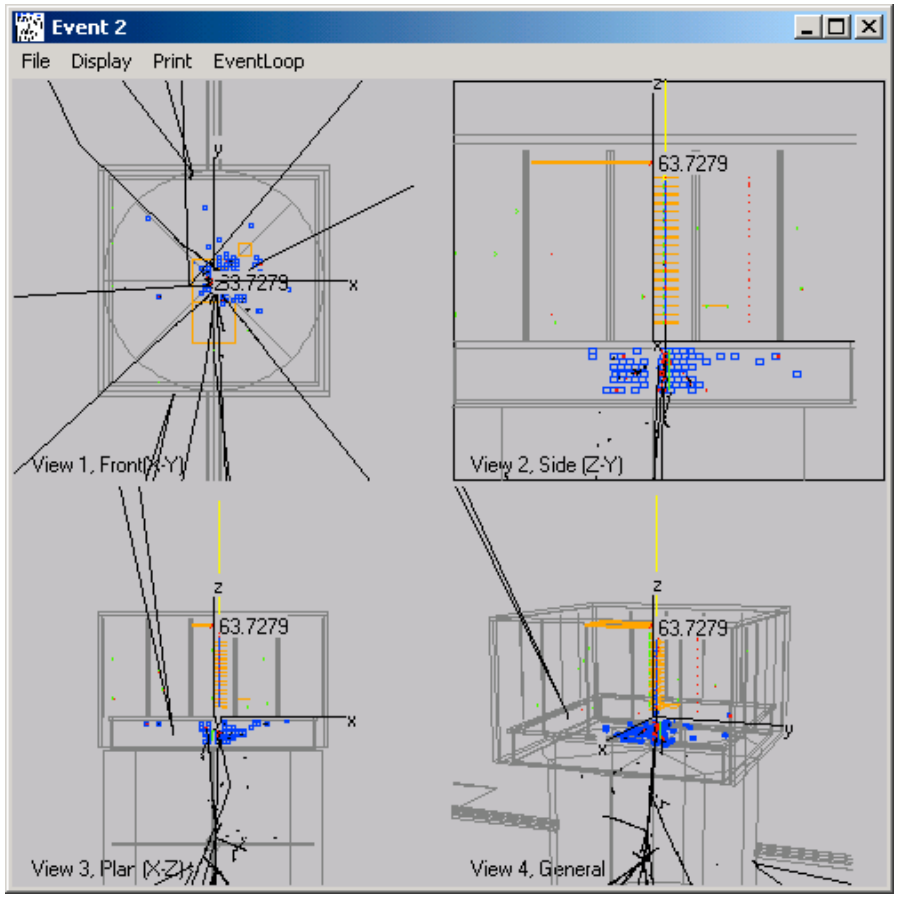
utilities

vcmt – gui for package manipulation



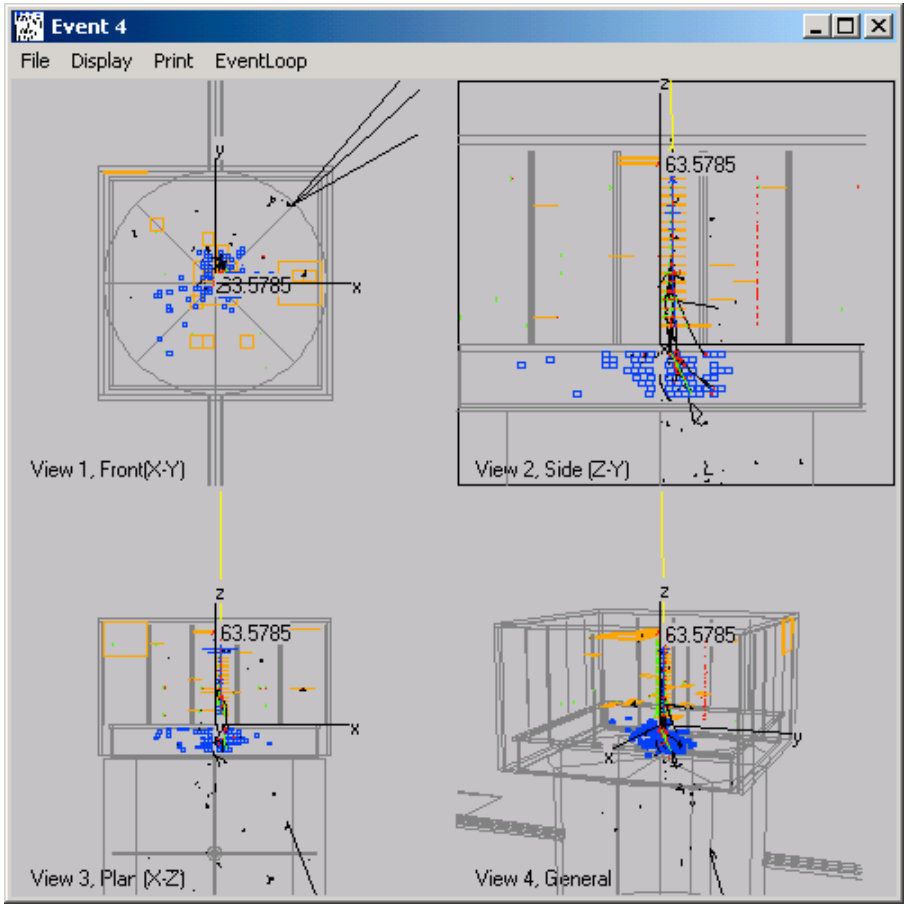
Available on windows and linux.
Glastpack available for real men.

Event Displays



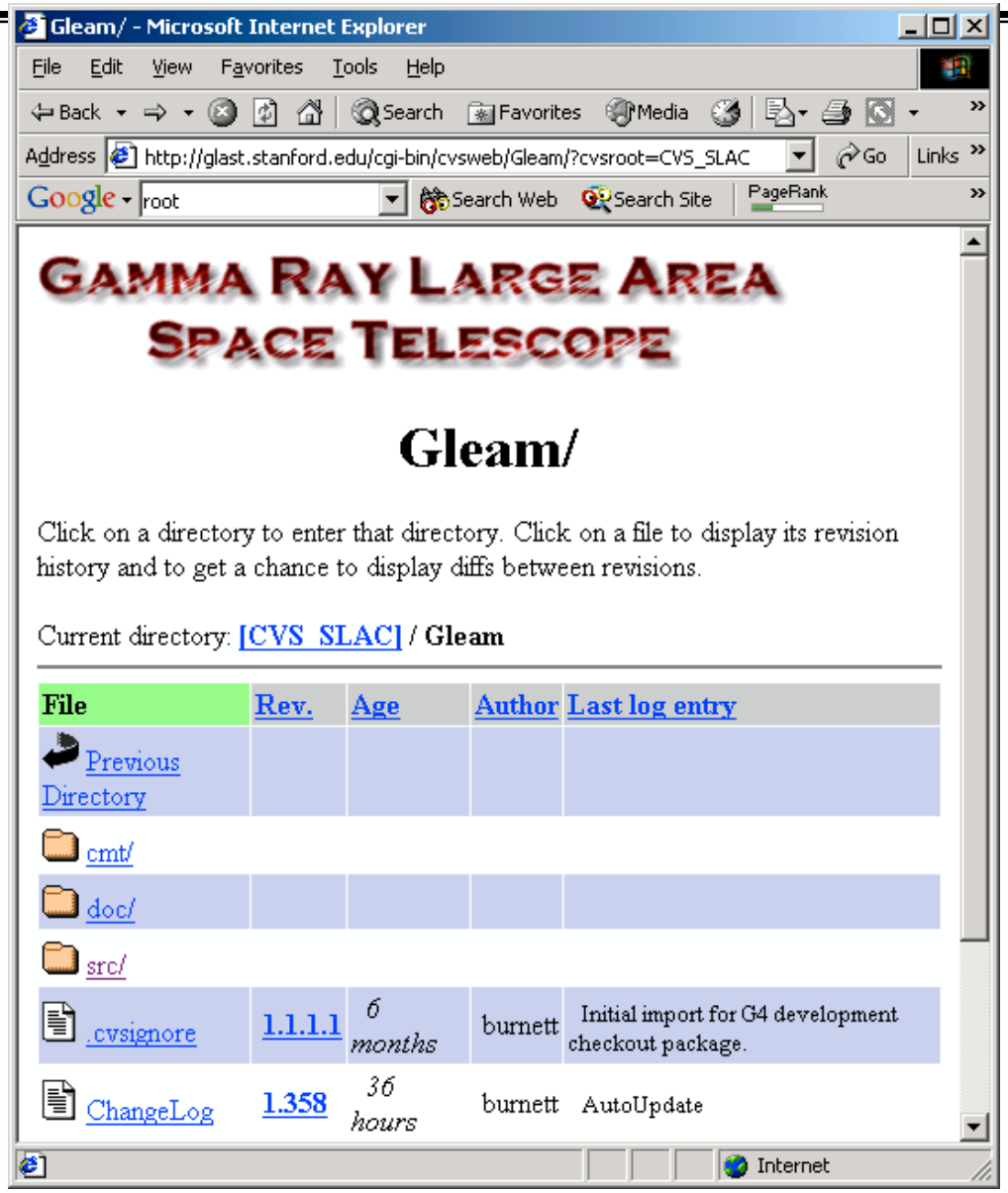
10 GeV proton

1 GeV gamma



FRED & WIRED coming soon as more interactive displays.

cvs web – peruse the code



Where to Get and Run the Code

- **Developer**
  - From cvs at SLAC
  - cvsroot = centaurusa.slac.stanford.edu
- **User**
  - Get binary distributions from our ftp site
    - Will be ready with Gleam v3
- **Run the code**
  - Your own box
    - You are the sysadmin
  - Prebuilt
    - SLAC linux farm
    - UW Windows terminal server
    - Releases already built

When? End of October for Gleam v3 !

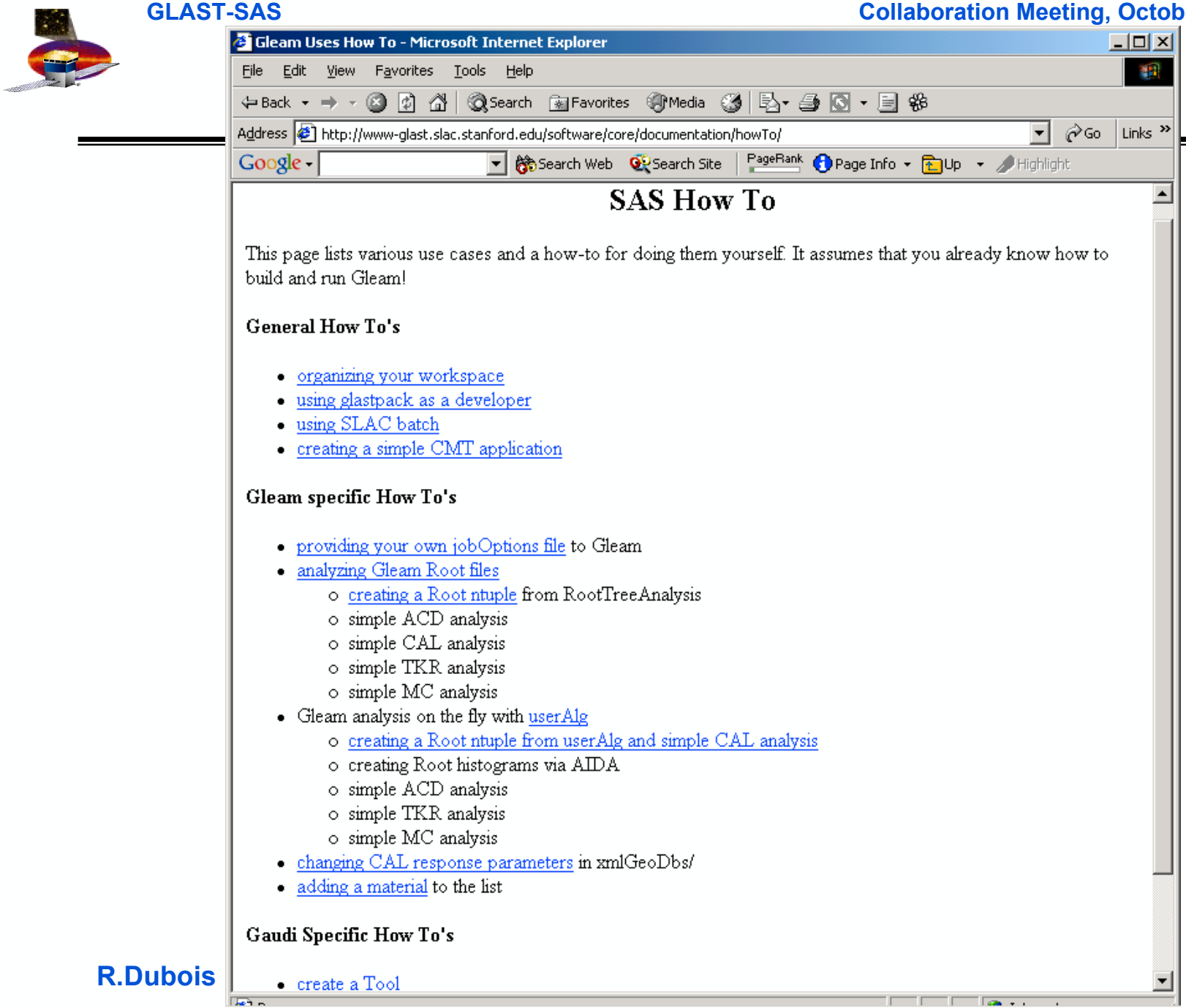
How to Run the Code

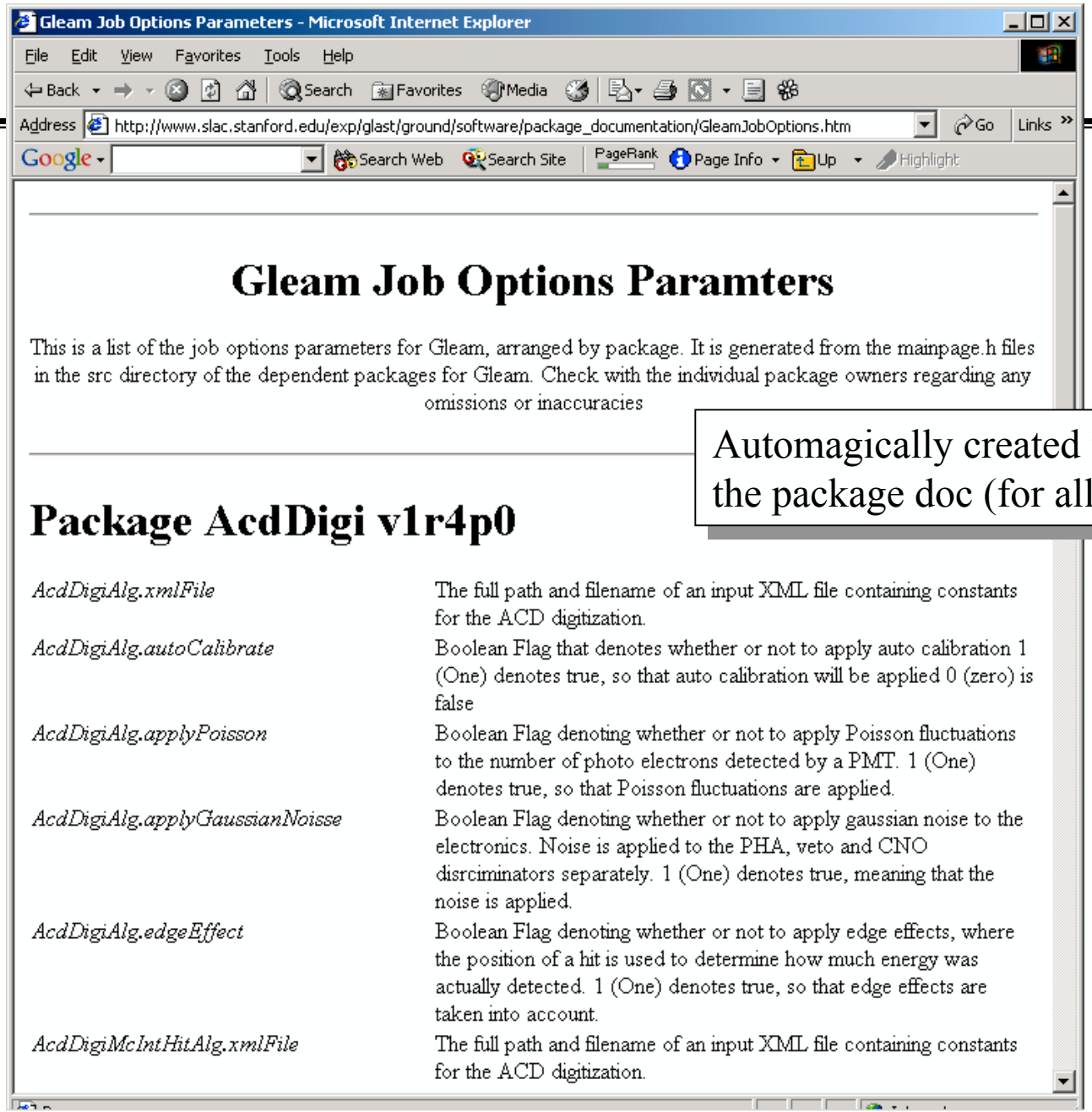
- Gleam is configurable from a "jobOptions" file:
  - Algorithms are loaded dynamically upon request
  - Can specify the exact sequence of algorithms to run, eg
    - Source + G4 + Digi + Recon creating output Root files
    - Recon from input Digi file, creating output Root file
    - Digi from input MC file, no output
  - Analyze results from output files (RootTreeAnalysis) or add your own analysis algorithm (userAlg) to Gleam
    - Analyze full TDS or Root trees or use the analysis ntuple and create/add your own ntuple columns

- See the Gleam Users Guide (aka RTFM)

  http://www-glast.slac.stanford.edu/software/gleam/userGuide/

  - Descriptions of obtaining and running the code
  - A growing Cookbook of How-To examples
  - mailto:helpsoftlist@glast.stanford.edu
  - Approaching critical mass for community of users who can answer queries


Gleam Uses How To - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www-glast.slac.stanford.edu/software/core/documentation/howTo/
Go Links
Google Search Web Search Site PageRank Page Info Up Highlight
SAS How To
This page lists various use cases and a how-to for doing them yourself. It assumes that you already know how to build and run Gleam!
General How To's
organizing your workspace
using glastpack as a developer
using SLAC batch
creating a simple CMT application
Gleam specific How To's
providing your own jobOptions file to Gleam
analyzing Gleam Root files
creating a Root ntuple from RootTreeAnalysis
simple ACD analysis
simple CAL analysis
simple TKR analysis
simple MC analysis
Gleam analysis on the fly with userAlg
creating a Root ntuple from userAlg and simple CAL analysis
creating Root histograms via AIDA
simple ACD analysis
simple TKR analysis
simple MC analysis
changing CAL response parameters in xmlGeoDbs/
adding a material to the list
Gaudi Specific How To's
create a Tool



Automagically created from the package doc (for all packages).

What's Coming in Gleam

- Continual improvement of subsystem digis and recons
  - ACD
    - Ribbons, efficiency maps
    - Calibrations
  - CAL
    - Revised leakage corrections
    - Non-linear tapers
    - Failure modes
    - Calibratrions
  - TKR
    - Further development in all areas – digi, patrec, fitting
    - Alignment algorithms
- Need to add Event Interpretation step
  - Make use of all the subsystem info
- Fancier event display
  - Separated off from Gleam proper
  - Expected in November (after Riccardo's teaching is done)
- Python interface available from Gaudi-central – starting to play with it
- Have NOT yet had the manpower to create a gui for Root analysis

Development Process

- Have been on 6-month cycle for major releases
  - Seems about right so far
  - Backward-incompatible upgrades are done!

- BBS (Beg Borrow & Steal) approach
  - Upside is all the features we gain for 'free', supplied and supported by others
  - Downside is keeping up with new versions
    - Major suspects for upgrade usually are
      - G4, Root, Gaudi
      - CMT
      - Compilers – interested in gcc 3.xx, and Studio 7 now
        - » These are essentially ANSI standard now; a good idea for us to upgrade
    - We expect to embark on these upgrades forthwith

Random Quotes from Users

- **Julie Mcenery – new to us from Goddard:**
  - **“I am stunned by the relative ease with which I could get things running. I was expecting more of a struggle.”**
- **Berrie Giebels**
  - **“It is easy now and there are more and more people available for help. Running it has also few surprises.. the only comments I got at the lab was that the event display was ugly. I think that it's a great tool and it only needs more "cookbooks", because some people will use it like me with a partial understanding of c++, but knowing a bit about Root AND what you want to look for is enough to actually use it.”**
- **It works! Have fun with it!**